Curso: Teoría Computacional - Grupo: T35A

Profesor: Juan Carlos González Ibarra

[juancarlos.gonzalez@upslp.edu.mx](mailto:juancarlos.gonzalez@upslp.edu.mx)

[teoriacomputacional@upslp.edu.mx](mailto:teoriacomputacional@upslp.edu.mx)

**Presentación**

Este es un curso diseñado para enseñarle los fundamentos matemáticos de la computación, junto con las técnicas necesarias para razonar sobre las estructuras que aparecen a lo largo de la informática. En consecuencia, las tareas de este curso están diseñadas para darle la oportunidad de jugar con el material y mejorar sus habilidades con pruebas matemáticas, teoría de la computabilidad y teoría de la complejidad.

**Propósitos**

* + Identificar que la teoría computacional es la columna vertebral de las matemáticas computacionales.

* + Plantear y resolver problemas matemáticos, así como modelos que te permitan interpretar lo que sucede en la teoría computacional.

**Competencias por desarrollar**

Comprender los conceptos matemáticos del comportamiento de una computadora, a través del estudio de los modelos computacionales: máquinas de Turing, autómatas finitos, autómatas de pila, gramáticas y lenguajes formales, que permitan introducirse en el conocimiento de la complejidad computacional.

**Competencia general**

Comprender la base teórica para la construcción de sistemas formales y utilizar técnicas de programación para modelarlos.

**Competencias específicas**

**Unidad 1 Teoría de conjuntos y Lenguajes Regulares**

Utiliza teoría de conjuntos para resolver problemas de distintas áreas de la computación.

* + Introducción
  + Revisión de los conceptos de matemáticas discretas
  + Autómata finito determinista
  + Autómata finito no determinista.
  + Expresiones regulares.
  + Lenguajes regulares
  + Lenguajes no regulares

#### 

#### Resultados de Aprendizaje

#### · El alumno recordara los conceptos y operaciones de conjuntos y técnicas de demostración que son necesarios para el curso.

#### · Se revisará la notación matemática necesaria.

#### · Finalmente, se abordarán los temas de Lenguajes y autómatas.

**Unidad 2 Lenguajes libre de contexto y La tesis de Church-Turing**

Emplea lenguajes formales para resolver problemas de comunicación entre los diferentes dispositivos dentro de un lenguaje de programación.

* + Lenguajes Libres de Contexto
  + Gramáticas libres de contexto
  + Autómatas de pila
  + Lenguajes no libres de contexto
  + La Tesis de Church-Turing
  + Máquinas de Turing
  + Variaciones de la máquina de Turing
  + Definición de algoritmo

**Resultados de Aprendizaje**

* + Analizar lenguajes libres de contexto.
  + Diseñará o construirá con lenguaje de programación Maquina de Turing.

**Unidad 3 Decibilidad, Reducciones, Introducción a la Complejidad Computacional**

Utiliza análisis de complejidad computacional para resolver problemas en la implementación de algoritmos computacionales.

* + Lenguajes decidibles
  + Problema del paro
  + Problemas no decidibles
  + Reducción de Turing
  + Complejidad
  + La clase polinomial
  + La clase no polinomial
  + Problemas no polinomiales completos

**Resultados de Aprendizaje**

Analizar algoritmos para su complejidad computacional.

Metodología de trabajo

03 September 2020

15:11

Para la asignatura de se utilizará como metodología de trabajo el Aprendizaje Basado en Problemas (ABP), por lo que frecuentemente te enfrentarás a situaciones que deberás resolver a partir de lo que has aprendido en la asignatura. La finalidad de la asignatura no sólo es conceptual, si no que la información sea utilizada o aplicada para la solución de problemas, para el logro de la competencia es fundamental seguir el cumplimiento cabal de cada una de las actividades planteadas, la ejercitación de procedimientos matemáticos o ejercicios prácticos, así como el constante estudio de los conceptos que forman parte de la asignatura. A continuación, se describen de forma general las estrategias metodológicas de enseñanza-aprendizaje. Para llevar a cabo un análisis de los problemas planteados se realizarán diversas actividades, algunas de ellas en el foro, las cuales deberás discutir con tus compañeros(as), con el fin de enriquecer tu aprendizaje, ya que te permitirán conocer otros puntos de vista y tomar en cuenta cosas que tú no harías. Estas actividades son formativas y para algunas utilizarás la herramienta de Tareas para poder enviarlas a tu Docente en línea.

**Foro de dudas**

En él podrás plasmar todas las inquietudes y cuestionamientos que te vayan surgiendo al momento de consultar los contenidos (materiales por unidad), así como al realizar las actividades y evidencia de aprendizaje. Tu docente en línea también podrá realizarte un diagnóstico de todos los conocimientos, relacionados con la asignatura, con los que cuentas, o bien, organizar equipos de trabajo si se requiere realizar una actividad en equipo con tus compañeros(as).

**Avisos del docente en línea.**

Este espacio fue diseñado para que el docente en línea pueda plasmar y comunicar tanto las actividades determinadas para esta asignatura como las complementarias; éstas últimas te aportarán elementos para alcanzar la competencia específica, es decir, tu objetivo por unidad. Es importante mencionar que deberás estar al pendiente de este espacio, porque el docente en línea puede comunicarse contigo y atender contingencias o problemáticas que vayan surgiendo en el semestre. Asimismo, te comunicará el diseño de cada una de las actividades que contribuirán a tu aprendizaje y asignar fechas de entrega, y que finalmente autogestiones tú tiempo requerido para esta asignatura, otra de las funciones de este espacio es que también te puede enviar material extra de consulta.

**Autorreflexiones**

Por otro lado, cuentas con la actividad de Autorreflexiones, en la cual tu docente en línea te formulará preguntas detonadoras para generarte una reflexión respecto a lo revisado en cada unidad, reflexión que podrás plasmar a través de un documento que, a su vez, podrás subir en la herramienta de tarea con el mismo nombre.

**Asignación a cargo del docente en línea**

Asimismo, cuentas con la pestaña de Asignación a cargo del docente en línea, en la cual podrás encontrar, debidamente configuradas, herramientas de tareas. En estas herramientas deberás subir las respuestas de las actividades complementarias determinadas y comunicadas por tu docente en línea mediante el espacio de Avisos del docente en línea, y te permitirán abarcar conocimientos y habilidades para alcanzar las competencias establecidas en la asignatura.

**Contenidos**

Por último, no olvides consultar los contenidos que fueron seleccionados, determinados y desarrollados por un equipo docente para cada unidad, ya que estos contenidos son el conocimiento mínimo que debes aprender para poder realizar las actividades mencionadas anteriormente y así concluir con éxito la asignatura de Teoría Computacional. ¡No dejes de hacerlo!

**Evaluación**

Para acreditar la asignatura se espera la participación responsable y activa del estudiante, contando con el acompañamiento y comunicación estrecha con su docente en línea, quien, a través de la retroalimentación permanente, podrá evaluar de manera objetiva su desempeño. En este contexto, la retroalimentación permanente es fundamental para promover el aprendizaje significativo y reconocer el esfuerzo. Es requisito indispensable la entrega oportuna de cada una de las tareas, actividades y evidencias, así como la participación en foros y demás actividades programadas en cada una de las unidades y conforme a las indicaciones dadas. Las rúbricas establecidas para cada actividad contienen los criterios y lineamientos para realizarlas, por lo que es importante que el (la) estudiante las revise antes de elaborar sus actividades. A continuación, se presenta el esquema de evaluación.

**Esquema de evaluación por parcial:**

Tareas 40%

Programación de ejercicios 30%

Participación en foros 20%

Examen parcial 10%

**Evaluación final:**

E-portafolio 20%

Proyecto final 100%

**Fuentes de consulta**

* Theory of Computation, George Tourlakis 27 April 2012 ISBN:9781118014783 <https://onlinelibrary.wiley.com/doi/book/10.1002/9781118315361>
* Introduction to the Theory of Computation. First Edition, Michael Sipser, PWS Publishing Company (1997)
* Teoría de Autómata y Lenguajes Formales: Dean Kelley. Prentice Hall (1998)
* Elements of the Theory of Computation: Harry Lewis, Cristos Papadimitriou, Prentice Hall (1997)
* Computers and Intractability: A Guide to the Theory of NP-Completeness. W H: Michael Garey y David Johnson. Freeman & Co.(1979)
* Lenguajes Formales: y teoria de la computacion: John Marti. Mc Graw Hill (2004)

**Foro de dudas:**

En este espacio podrán hacerme saber sus dudas, preguntas, comentarios y sugerencias académicas respecto a los contenidos, trabajos y actividades de la primera unidad.

Actividades

03 September 2020

15:12

**Instrucciones**

* 1. Consulta la *Planeación del docente en línea*.
  2. Consulta los criterios de evaluación.
  3. Revisa los recursos proporcionados por tu docente en línea.
  4. Lee el contenido de la unidad.
  5. Para revisar la notación y las definiciones, lea el resumen de "Basics" publicado en el sitio web de la clase y también lea el capítulo inicial (Capítulo 0) del libro de texto Sipser.
  6. Identifica la actividad que realizarás.
  7. Verifica la fecha y los requisitos de entrega.
  8. Envíe sus soluciones a través del sistema de blackboard (el sistema en línea simplemente se negará a aceptar envíos tardíos después de la fecha límite, por lo que, literalmente, ni siquiera podrá entregarlos tarde incluso si lo intenta).
  9. Las tareas vencen antes de las 11:59 pm de cada viernes; no se aceptarán presentaciones tardías.
  10. Guarda tu actividad con la nomenclatura indicada.
  11. Carga en este espacio el archivo con la actividad.
  12. Ajusta lo que se te indique y carga tu archivo de nuevo.
  13. En el raro caso de que eso no sea posible enviarlo a través de blackboard, envíe un correo electrónico a [teoriacomputacional@upslp.edu.mx](mailto:teoriacomputacional@upslp.edu.mx).
  14. Verifica que tu documento no pese más de 5 MB.
  15. Los argumentos informales son aceptables, pero hágalos lo suficientemente precisos / detallados / convincentes para que sean rigurosos.
  16. Por favor, asigne un nombre único a su archivo PDF enviado con su propio nombre y su ID de carrera, usando el siguiente formato de nombre de archivo: Apellido\_PrimerNombre\_ComputingID\_Homework\_1.pdf (por ejemplo, "Gonzalez\_Carlos\_180864\_Homework\_1.pdf").
  17. Espera la retroalimentación de tu docente en línea.
  18. Por favor, no posponga las cosas ni se apresure, ya que no le funcionará bien en este curso.
  19. Por favor, no nos ponga (ni a usted mismo) en una posición incómoda en la que nos obligue a decir “te lo dijimos”.
  20. Entonces, si no ha terminado la tarea antes de la fecha límite, su mejor estrategia es simplemente entregar la parte que ha terminado para ese momento (en lugar de llegar tarde y no recibir ningún crédito por esa tarea), y luego ir reenviar por favor asegúrese de que no se demore en entregar las asignaciones futuras.
  21. Tenga en cuenta que, si planea retrasar la fecha límite e intenta entregar una tarea unos minutos antes de la fecha límite, y algo falla en el sistema (por ejemplo, retrasos en la red o problemas con el servidor u otros problemas logísticos), eso también será su responsabilidad ya que eligió adelantar el plazo e ignorar nuestras instrucciones para no posponer las cosas. En tal caso, habría perdido la oportunidad de entregar la tarea actual. Por lo tanto, su mejor estrategia en general es entregar una tarea varios días antes de su fecha límite (estos posibles escenarios de fallas ya están “incorporados” a las muy generosas fechas límite).
  22. Por favor, no nos diga después del hecho de que no se dio cuenta de todo esto, o que decidió no planificar en consecuencia, porque entonces le recordaremos que vuelva a leer estas instrucciones y consejos aquí, y también le recordaremos que no recibir crédito por una tarea es un precio muy pequeño para pagar por una lección de vida tan importante.

**Tareas 40 %**

**Utiliza el témplate de tarea para realizar los ejercicios de cada unidad.**

1. Lea atentamente la Política de trampas en el programa del curso.
2. No envíe respuestas que no comprenda completamente (nos reservamos el derecho de pedirle que explique verbalmente en persona y en persona, y definitivamente ejerceremos esta opción).
3. Resuelva tantos problemas como pueda; explique / pruebe todas las respuestas.
4. Indique claramente la respuesta corta / idea de prueba primero, y luego su respuesta / prueba completa.
5. Envíe solo las páginas provistas (use más hojas solo si es absolutamente necesario).
6. Edite directamente este archivo Word de tarea, inserte sus respuestas allí y envíe su tarea completa como un archivo PDF adjunto en la página web del curso “Plantilla para tareas”. Por favor asegúrese de poner su nombre en la primera página y firme la promesa de que cumplió con el Código de Honor.

**Programación 30%**

**Utiliza el témplate de programación para realizar la documentación de cada código fuente.**

**Instrucciones:**

1. Programar en el lenguaje de programación asignado
2. Los programas están ubicados en la carpeta (source\_code).
3. El código se debe compilar en cualquier sistema operativo.
4. El código se sube en <https://github.com/upslp-teoriacomputacional>
5. Ejemplo. 00180864\_conjuntos. go
6. El programa se evaluará de acuerdo con la rúbrica que se anexa.
7. Las evidencias del programa desarrollado es la plantilla del código fuente comentado.
8. También se puede mostrar el funcionamiento de los programas en los inicios o términos de clase antes de la fecha límite.

**Participación – Foros**

**Utilizar la rúbrica de análisis de temas**

**Actividad colaborativa - Análisis del tema 20%**

1. Consulta la Planeación del docente en línea.
2. Lee el contenido de la unidad.
3. Responde la pregunta que formule el docente en línea (mediador del foro), de forma clara y asertiva.
4. Retroalimenta a tus compañeros(as) para enriquecer sus aportaciones.
5. Argumenta, de forma respetuosa y clara, tus comentarios.

**Examen 10%**

1. **Consulta** el espacio de *Planeación didáctica del docente en línea* para más indicaciones.
2. **Lee** detenidamente los planteamientos que se te presentan a continuación.
3. **Elije** la opción que consideres correcta.

\*Tendrás **dos** oportunidades para contestar el cuestionario. Se tomará en cuenta la calificación más alta.

**\*Recuerda** consultar el *Foro de dudas*, para resolver inquietudes que te surjan en la elaboración de esta actividad. También puedes aportar soluciones a cuestionamientos de tus compañeros(as).

Presentaciones

03 September 2020

15:16

**Unidad 1**

Chapter 1 Teoría de conjuntos

Chapter 2 Inducción y Deducción

Chapter 3 Autómata Finito

Chapter 4 Autómata Finito no determinístico

**Unidad 2**

Chapter 5 Lenguajes Regulares

Chapter 6 Programando Expresiones Regulares

Chapter 7 Lenguajes y Gramáticas Libre de Contexto

Chapter 8 Autómata de Pila

**Unidad 3**

Chaptert 9 Definición de Computabilidad

Chapter 10 Programando Maquina de Turing

Chapter 11 Complejidad Polinomial

Chapter 12 Máquinas de Turing no determinísticas

Chapter 13 Complejidad No Polinomial

Código de honor

03 September 2020

15:12

* 1. Las actividades entregables son tareas, participación, programas y otra actividad a cargo del docente en línea.
  2. Las actividades no entregables son cuestionarios, foros y preguntas de reflexión.
  3. El docente será encargado de indicar las fechas de3 entrega puntual y tardía.
  4. Cada estudiante tendrá dos intentos para enviar las actividades entregables siempre y cuando se realice con envió puntual, en envió tardío solo existe un intento.
  5. Las actividades entregables enviadas de forma tardía serán evaluadas con sanción, en una escala 50/100.
  6. En la entrega de participación en foros y cuestionarios no habrá prorrogas, se entregan en fecha indicada en planeación deberán responderse todos los planteamientos enunciados por el docente.
  7. Los estudiantes cuentan con una semana para entrega correcciones en envió puntual, a partir de las fechas de retroalimentación.
  8. Los estudiantes con entrega tardía no podrán hacer correcciones de las actividades evaluadas.
  9. Las actividades enviadas después de la fecha tardía, las que estén incompletas, las que correspondan a otras asignaturas o unidades, así como las actividades en blanco serán evaluadas con 0/100, no podrán entregarse correcciones de estas actividades.
  10. Únicamente se evaluarán actividades completas conforme a las instrucciones, a la unidas, a la asignatura y al formato solicitado.
  11. Las actividades con plagio serán evaluadas con 0/100, no podrán entregarse correcciones de esta actividad.
  12. Cada estudiante deberá consultar diariamente el correo para mantener contacto por medios oficiales.
  13. Puede trabajar e intercambiar ideas en grupos, pero no se permite la copia literal de las soluciones de otras personas.
  14. Reúnase a asesorías con frecuencia, haga preguntas con regularidad y asista a las sesiones semanales de resolución de problemas.
  15. \*Recuerda consultar el *Foro de dudas*, para resolver inquietudes que te surjan en la elaboración de esta actividad. También puedes aportar soluciones a cuestionamientos de tus compañeros(as).

FAQ: Class Notebook in Microsoft Teams

**Where can I get more Class Notebook questions answered?**

[OneNote Class Notebook help center](http://aka.ms/onenoteedufaq)

**Questions? Need assistance?**

File a support ticket at: [https://aka.ms/EDUSupport](https://aka.ms/FREEDUSupport)

**Where can I find OneNote and Class Notebook training resources?**

A few short interactive courses on the Microsoft Education Center:

* [OneNote Class Notebook: A teacher's all-in-one notebook for students](https://aka.ms/FRECNBTrainingLink1)
* [Getting Started with OneNote - Microsoft in Education](https://aka.ms/FRECNBTrainingLink2)
* [OneNote: your one-stop resource - Microsoft in Education](https://aka.ms/FRECNBTrainingLink3)

**Experiencing permissions issues with your Class Notebook?**

[Troubleshoot notebook permissions](https://aka.ms/FREFixPermissions)

**Where do I find the settings to manage my Class Notebook?**

Go into your Class Notebook in Teams and click the Class Notebook toolbar, and then click Manage Notebook. Teachers can edit sections, copy a notebook link, lock the Collaboration Space, or create a Teacher-only section group here.

To manage Collaboration Space permissions or generate parent and guardian links, open Class Notebook in OneNote and then select Manage Notebooks.

**Join the conversation on social media:**

Twitter: [@OneNoteEDU](http://twitter.com/onenoteedu) and [@msonenote](http://twitter.com/msonenote)

Facebook: [OneNote](https://www.facebook.com/OneNote)

Configuracion de GitHub

03 September 2020

16:17

<https://docs.github.com/es/github/authenticating-to-github/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

Fechas de examenes

05 September 2020

15:23

PRIMER PARCIAL

Aplicación de exámenes Del 6 de Octubre

SEGUNDO PARCIAL

Aplicación de exámenes Del 4 de Noviembre

TERCER PARCIAL

Aplicación de exámenes 1 de Dic

FINALES 10 de Diciembre

EXTRAORDINARIOS 15 Dic

Curso: Teoría Computacional - Grupo: T35A

Profesor: Juan Carlos González Ibarra

[juancarlos.gonzalez@upslp.edu.mx](mailto:juancarlos.gonzalez@upslp.edu.mx)

[teoriacomputacional@upslp.edu.mx](mailto:teoriacomputacional@upslp.edu.mx)

**Presentación**

Este es un curso diseñado para enseñarle los fundamentos matemáticos de la computación, junto con las técnicas necesarias para razonar sobre las estructuras que aparecen a lo largo de la informática. En consecuencia, las tareas de este curso están diseñadas para darle la oportunidad de jugar con el material y mejorar sus habilidades con pruebas matemáticas, teoría de la computabilidad y teoría de la complejidad.

**Propósitos**

* + Identificar que la teoría computacional es la columna vertebral de las matemáticas computacionales.

* + Plantear y resolver problemas matemáticos, así como modelos que te permitan interpretar lo que sucede en la teoría computacional.

**Competencias por desarrollar**

Comprender los conceptos matemáticos del comportamiento de una computadora, a través del estudio de los modelos computacionales: máquinas de Turing, autómatas finitos, autómatas de pila, gramáticas y lenguajes formales, que permitan introducirse en el conocimiento de la complejidad computacional.

**Competencia general**

Comprender la base teórica para la construcción de sistemas formales y utilizar técnicas de programación para modelarlos.

**Competencias específicas**

**Unidad 1 Teoría de conjuntos y Lenguajes Regulares**

Utiliza teoría de conjuntos para resolver problemas de distintas áreas de la computación.

* + Introducción
  + Revisión de los conceptos de matemáticas discretas
  + Autómata finito determinista
  + Autómata finito no determinista.
  + Expresiones regulares.
  + Lenguajes regulares
  + Lenguajes no regulares

#### 

#### Resultados de Aprendizaje

#### · El alumno recordara los conceptos y operaciones de conjuntos y técnicas de demostración que son necesarios para el curso.

#### · Se revisará la notación matemática necesaria.

#### · Finalmente, se abordarán los temas de Lenguajes y autómatas.

**Unidad 2 Lenguajes libre de contexto y La tesis de Church-Turing**

Emplea lenguajes formales para resolver problemas de comunicación entre los diferentes dispositivos dentro de un lenguaje de programación.

* + Lenguajes Libres de Contexto
  + Gramáticas libres de contexto
  + Autómatas de pila
  + Lenguajes no libres de contexto
  + La Tesis de Church-Turing
  + Máquinas de Turing
  + Variaciones de la máquina de Turing
  + Definición de algoritmo

**Resultados de Aprendizaje**

* + Analizar lenguajes libres de contexto.
  + Diseñará o construirá con lenguaje de programación Maquina de Turing.

**Unidad 3 Decibilidad, Reducciones, Introducción a la Complejidad Computacional**

Utiliza análisis de complejidad computacional para resolver problemas en la implementación de algoritmos computacionales.

* + Lenguajes decidibles
  + Problema del paro
  + Problemas no decidibles
  + Reducción de Turing
  + Complejidad
  + La clase polinomial
  + La clase no polinomial
  + Problemas no polinomiales completos

**Resultados de Aprendizaje**

Analizar algoritmos para su complejidad computacional.